

**IMPLEMENTASI FUZZING MENGGUNAKAN AFL++  
UNTUK MENEMUKAN KERENTANAN PADA PERANGKAT  
LUNAK BERBASIS BINARY**

**SKRIPSI**

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana  
Program Studi Teknik Komputer



disusun oleh

**AHMAD HASSAN RASYIDI**

**20.83.0577**

Kepada

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2025**

**IMPLEMENTASI FUZZING MENGGUNAKAN AFL++  
UNTUK MENEMUKAN KERENTANAN PADA PERANGKAT  
LUNAK BERBASIS BINARY  
SKRIPSI**

untuk memenuhi salah satu syarat mencapai derajat Sarjana  
Program Studi *Teknik Komputer*



disusun oleh  
**AHMAD HASSAN RASYIDI**  
**20.83.0577**

Kepada

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA  
2025**

**HALAMAN PERSETUJUAN**

**SKRIPSI**

**IMPLEMENTASI FUZZING MENGGUNAKAN AFL++ UNTUK  
MENEMUKAN KERENTANAN PADA PERANGKAT LUNAK BERBASIS  
BINARY**

yang disusun dan diajukan oleh

**Ahmad Hassan Rasyidi**

**20.83.0577**

telah disetujui oleh Dosen Pembimbing Skripsi  
pada tanggal 9 Juni 2025

Dosen Pembimbing,



**Muhammad Rudyanto Arief, S.T., M.T.**  
**NIK. 190302098**

**HALAMAN PENGESAHAN**  
**SKRIPSI**  
**IMPLEMENTASI FUZZING MENGGUNAKAN AFL++ UNTUK**  
**MENEMUKAN KERENTANAN PADA PERANGKAT LUNAK BERBASIS**  
**BINARY**

yang disusun dan diajukan oleh

**Ahmad Hassan Rasyidi**

**20.83.0577**

Telah dipertahankan di depan Dewan Penguji  
pada tanggal 30 Juni 2025

**Susunan Dewan Penguji**

**Nama Penguji**

**Tanda Tangan**

**Joko Dwi Santoso, M. Kom.**  
**NIK. 190302181**

**Ali Mustopa, S.Kom, M.Kom.**  
**NIK. 190302192**

**M. Rudyanto Arief, S.T., M.T.**  
**NIK. 190302098**



Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Komputer  
Tanggal 30 Juni 2025

**DEKAN FAKULTAS ILMU KOMPUTER**



**Prof. Dr. Kusrini, M.Kom.**  
**NIK. 190302106**

## HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan di bawah ini,

Nama mahasiswa : Ahmad Hassan Rasyidi  
NIM : 20.83.0577

Menyatakan bahwa Skripsi dengan judul berikut:

**Implementasi Fuzzing Menggunakan AFL++ Untuk Menemukan Kerentanan Pada Perangkat Lunak Berbasis Binary**

Dosen Pembimbing : Muhammad Rudyanto Arief, S.T., M.T.

1. Karya tulis ini adalah benar-benar **ASLI** dan **BELUM PERNAH** diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian **SAYA** sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab **SAYA**, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini **SAYA** buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka **SAYA** bersedia menerima **SANKSI AKADEMIK** dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 30 Juni 2025

Yang Menyatakan,



Ahmad Hassan Rasyidi



## KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyang, kami panjatkan puja dan puji syukur atas kehadiran-Nya, yang telah melimpahkan rahmat, serta hidayah, dan inayah-Nya sehingga saya dapat menyelesaikan penyusunan skripsi ini dengan judul “Implementasi Fuzzing menggunakan AFL++ untuk menemukan kerentanan pada perangkat lunak berbasis binary”, Sholawat serta salam semoga tetap tercurahkan kepada junjungan Nabi Muhammad SAW yang senantiasa kita harapkan syafaatnya di hari akhir kelak.

Skripsi ini penulis ajukan untuk memenuhi syarat kelulusan pada mata kuliah Skripsi di Fakultas Ilmu Komputer, Program Studi Teknik Komputer, Universitas Amikom Yogyakarta. Adapun penyusunan skripsi ini juga merupakan salah satu syarat untuk memperoleh gelar Strata 1. Dalam penyusunan skripsi ini berbagai pihak telah memberikan dorongan, bantuan, serta masukan sehingga dalam kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Ayah serta Ibu penulis yang senantiasa memberikan dukungan, do’a, dan motivasi untuk penulis sehingga dapat menyelesaikan pendidikan di Universitas Amikom Yogyakarta
2. Bapak Prof. Dr. M. Suyanto, M.M, selaku Rektor Universitas Amikom
3. Ibu Prof. Dr. Kusri, M. Kom. selaku Dekan Fakultas Ilmu Komputer Universitas Amikom Yogyakarta.
4. Bapak Dr. Dony Ariyus, M.Kom selaku Kepala Prodi Teknik Komputer.
5. Bapak M. Rudyanto Arief, S.T, M.T selaku dosen pembimbing yang dengan sabar dan ikhlas telah meluangkan waktu dan memberikan ilmunya kepada penulis dalam memberikan petunjuk dan bimbingan sehingga penulis dapat menyelesaikan skripsi ini.
6. Kepada seseorang yang tidak kalah penting kehadirannya, Fravangastha Hangyang Levi Garniswaninghyun. Terima kasih telah menjadi bagian

dari proses perjalanan penulis untuk menyelesaikan skripsi. Berkontribusi baik tenaga, waktu, mendukung serta menghibur penulis dalam kesedihan, mendengarkan keluh kesah serta meyakinkan penulis untuk pantang menyerah sampai skripsi ini terselesaikan.

7. Rangga Wahyu Setiawan dan Adam Firdaus selaku teman dekat penulis yang selalu memberikan semangat agar skripsi saya dapat segera diselesaikan.
8. Semua pihak yang telah memberikan kontribusi, baik secara langsung maupun tidak langsung yang tidak dapat saya sebutkan satu per satu dalam penyelesaian skripsi ini. Terima kasih atas kesempatan dan dukungan yang telah diberikan selama proses penulisan skripsi ini. Semoga skripsi ini dapat memberikan manfaat dan kontribusi yang berarti bagi perkembangan ilmu pengetahuan di masa depan yang akan datang.

Yogyakarta, 24 Juni 2025

Penulis

Ahmad Hassan Rasyidi

## DAFTAR ISI

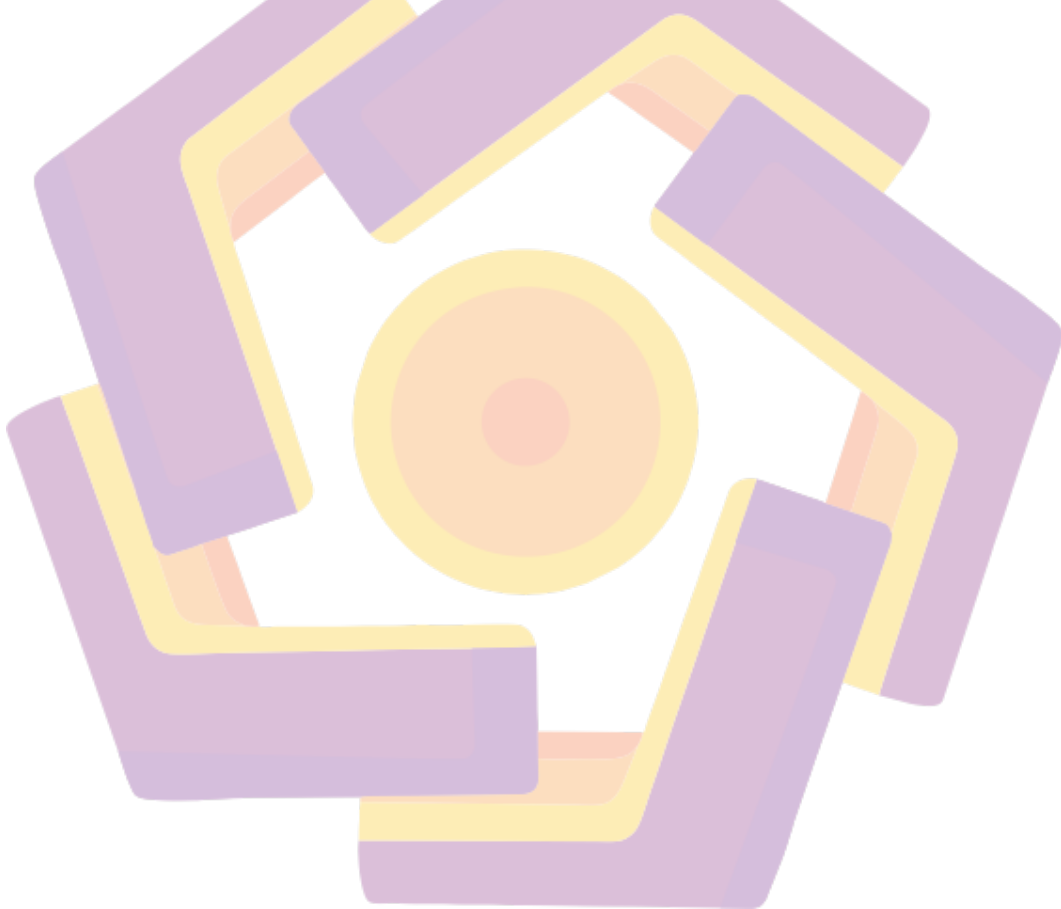
|                                                       |     |
|-------------------------------------------------------|-----|
| HALAMAN JUDUL.....                                    | i   |
| HALAMAN PERSETUJUAN.....                              | ii  |
| HALAMAN PENGESAHAN.....                               | iii |
| HALAMAN PERNYATAAN KEASLIAN SKRIPSI.....              | iv  |
| DAFTAR ISI.....                                       | vii |
| DAFTAR TABEL.....                                     | xi  |
| DAFTAR GAMBAR.....                                    | x   |
| DAFTAR ISTILAH.....                                   | xiv |
| INTISARI.....                                         | xv  |
| ABSTRACT.....                                         | xvi |
| <b>BAB I</b>                                          |     |
| PENDAHULUAN.....                                      | 1   |
| 1.1 Latar Belakang.....                               | 1   |
| 1.2 Rumusan Masalah.....                              | 3   |
| 1.3 Batasan Masalah.....                              | 3   |
| 1.4 Tujuan Penelitian.....                            | 4   |
| 1.5 Manfaat Penelitian.....                           | 4   |
| 1.6 Sistematika Penulisan.....                        | 4   |
| <b>BAB II</b>                                         |     |
| TINJAUAN PUSTAKA.....                                 | 6   |
| 2.1 Studi Literatur.....                              | 6   |
| 2.3 Dasar Teori.....                                  | 14  |
| 2.3.1 Security Assessment.....                        | 14  |
| 2.3.2 Tujuan Security Assessment.....                 | 15  |
| 2.3.3 Jenis-Jenis Security Assessment.....            | 16  |
| 2.3.4 Vulnerability Assessment.....                   | 16  |
| 2.3.5 Penetration Testing.....                        | 16  |
| 2.3.6 Fuzzing.....                                    | 17  |
| 2.3.7 AFL++ (American Fuzzy Lop).....                 | 18  |
| 2.3.8 Fitur Utama AFL++.....                          | 20  |
| 2.3.9 Kerentanan Perangkat Lunak Berbasis Binary..... | 22  |
| 2.3.10 Teknik Fuzzing dengan AFL++.....               | 23  |
| <b>BAB III</b>                                        |     |
| METODE PENELITIAN.....                                | 27  |
| 3.1 Objek Penelitian.....                             | 27  |



|                                                    |           |
|----------------------------------------------------|-----------|
| 3.2 Alur Penelitian.....                           | 27        |
| 3.2.1 Persiapan Lingkungan.....                    | 29        |
| 3.2.2 Pelaksanaan Fuzzing.....                     | 31        |
| 3.2.3 Analisis Data.....                           | 31        |
| 3.3 Alat dan Bahan.....                            | 32        |
| 3.3.1 Perangkat Keras.....                         | 33        |
| 3.3.2 Perangkat Lunak.....                         | 33        |
| 3.3.3 Bahan Fuzzing.....                           | 34        |
| <b>BAB IV</b>                                      |           |
| <b>HASIL DAN PEMBAHASAN.....</b>                   | <b>36</b> |
| 4.1 Hasil.....                                     | 36        |
| 4.1.1 PDFCrack.....                                | 36        |
| 4.1.2 Dataset Input Awal / Seed Corpus.....        | 40        |
| 4.1.3 Sumber Dataset.....                          | 41        |
| 4.1.4 Distribusi Berdasarkan Versi PDF.....        | 42        |
| 4.1.5 Proses Pembuatan Dataset.....                | 42        |
| 4.1.6 Preprocessing Dataset untuk Fuzzing AFL..... | 57        |
| 4.1.7 Pelaksanaan Fuzzing.....                     | 64        |
| 4.1.8 Analisis Data.....                           | 71        |
| 4.2 Pembahasan.....                                | 79        |
| <b>BAB V</b>                                       |           |
| <b>PENUTUP.....</b>                                | <b>82</b> |
| 5.1 Kesimpulan.....                                | 82        |
| 5.2 Saran.....                                     | 82        |
| <b>REFERENSI.....</b>                              | <b>84</b> |

## DAFTAR TABEL

|                                                                                                    |    |
|----------------------------------------------------------------------------------------------------|----|
| Tabel 2.1. merupakan detail penelitian terdahulu yang masih terkait dengan tema yang penulis kaji. | 7  |
| Tabel 3.1. Komposisi Dataset Input Awal Berdasarkan Jenis PDF                                      | 27 |
| Tabel 4.1. Spesifikasi Perangkat Lunak                                                             | 34 |
| Tabel 4.2. Karakteristik Library dalam Pembuatan Dataset                                           | 39 |
| Tabel 4.3. Distribusi Dataset Berdasarkan Versi PDF                                                | 40 |
| Tabel 4.4. Hasil Pengujian Fuzzing – Statistik dan Analisis Kerentanan                             | 78 |



## DAFTAR GAMBAR

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| Gambar 2.1. Skema Diagram                                                   | 10 |
| Gambar 2.2. Skema Diagram Alir                                              | 11 |
| Gambar 2.1. Kerangka Konsep                                                 | 13 |
| Gambar 3.1. Flowchart Alur Penelitian                                       | 26 |
| Gambar 4.1. Tampilan Website Repository Kali Linux Perangkat Lunak Pdferack | 36 |
| Gambar 4.2. Proses Cloning Perangkat Lunak Pdferack                         | 36 |
| Gambar 4.3. Kode Sumber Pdferack                                            | 37 |
| Gambar 4.4. Proses Kompilasi Perangkat Lunak menggunakan GCC                | 38 |
| Gambar 4.5. Opsi-opsi pada perangkat lunak pdferack                         | 38 |
| Gambar 4.6. Import Library Python                                           | 41 |
| Gambar 4.7. Pendefinisian variabel output PDF                               | 41 |
| Gambar 4.8. Pendefinisian variabel versi PDF                                | 41 |
| Gambar 4.9. Fungsi pembuatan PDF sederhana                                  | 42 |
| Gambar 4.10. Fungsi pembuatan PDF dengan banyak object                      | 42 |
| Gambar 4.11. Fungsi pembuatan PDF dengan embedded font                      | 43 |
| Gambar 4.12. Fungsi pembuatan PDF dengan form                               | 43 |
| Gambar 4.13. Fungsi pembuatan PDF terpassword                               | 44 |
| Gambar 4.14. Fungsi pembuatan PDF disertai Error                            | 45 |
| Gambar 4.15. Fungsi pembuatan PDF ukuran besar atau kecil                   | 45 |
| Gambar 4.16. Fungsi pembuatan PDF yang XREF corrupt                         | 46 |
| Gambar 4.17. Fungsi pembuatan PDF dengan Trailer yang tidak valid           | 46 |
| Gambar 4.18. Fungsi pembuatan PDF dengan Stream yang tidak valid            | 47 |
| Gambar 4.19. Fungsi pembuatan PDF disertai Orphaned Object                  | 47 |
| Gambar 4.20. Fungsi pembuatan PDF dengan beberapa Startxref                 | 48 |
| Gambar 4.21. Fungsi pembuatan PDF dengan disertai Indirect Loop             | 49 |
| Gambar 4.22. Pemanggilan semua fungsi yang didefinisikan sebelumnya         | 49 |
| Gambar 4.23. Output Folder PDF                                              | 50 |
| Gambar 4.24. Output dari PDF 1.0                                            | 51 |
| Gambar 4.25. Output dari PDF 1.1                                            | 51 |
| Gambar 4.26. Output dari PDF 1.2                                            | 52 |
| Gambar 4.27. Output dari PDF 1.3                                            | 52 |
| Gambar 4.28. Output dari PDF 1.4                                            | 53 |
| Gambar 4.29. Output dari PDF 1.5                                            | 53 |
| Gambar 4.30. Output dari PDF 1.6                                            | 54 |

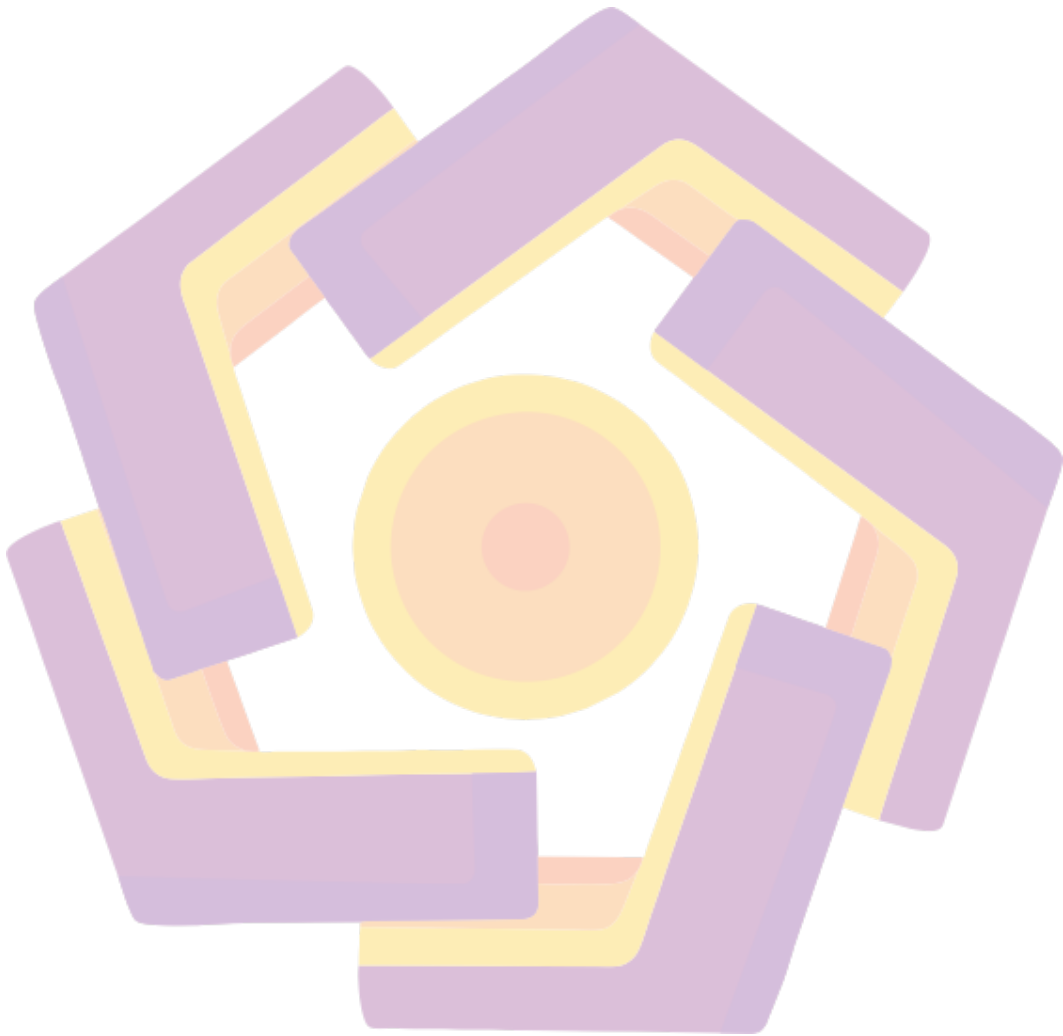
|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Gambar 4.31. Output dari PDF 1.7                                                              | 54 |
| Gambar 4.32. Output dari PDF 2.0                                                              | 55 |
| Gambar 4.33. Kode Sumber PDFCrack                                                             | 56 |
| Gambar 4.34. Proses Compiling Perangkat Lunak dengan Instrumentasi                            | 57 |
| Gambar 4.35. Validasi Instrumentasi telah terpasang pada Perangkat Lunak                      | 57 |
| Gambar 4.36. Proses Minimalisasi Input Seed Corpus                                            | 58 |
| Gambar 4.37. Output hasil Minimalisasi Input Seed Corpus                                      | 58 |
| Gambar 4.38. Validasi Input Seed Corpus yang telah terminimalisasi dengan afl-fuzz            | 60 |
| Gambar 4.39. Penyebab Program Tidak Berjalan                                                  | 60 |
| Gambar 4.40. Proses penghapusan Input Seed Corpus yang membuat perangkat lunak tidak berjalan | 61 |
| Gambar 4.41. Hasil akhir Input Seed Corpus setelah dilakukan Minimalisasi                     | 61 |
| Gambar 4.42. Perangkat Lunak berjalan dengan lancar dengan afl-fuzz                           | 62 |
| Gambar 4.43. Proses Kompilasi Perangkat Lunak dengan AddressSanitizer                         | 63 |
| Gambar 4.45. Skrip Automatisasi Fuzzing untuk melakukan Parallel Fuzzing                      | 64 |
| Gambar 4.46. Validasi Proses Fuzzing berjalan dengan Lancar                                   | 64 |
| Gambar 4.47. Proses Logging menggunakan afl-whatsup                                           | 65 |
| Gambar 4.48. Tampilan AFL++ pada fuzzer-pdfcrack03                                            | 66 |
| Gambar 4.49. Tampilan AFL++ pada fuzzer-pdfcrack02                                            | 66 |
| Gambar 4.50. Tampilan AFL++ pada fuzzer-pdfcrack01                                            | 67 |
| Gambar 4.51. Output Crash pada fuzzer-pdfcrack01                                              | 67 |
| Gambar 4.52. Output Crash pada fuzzer-pdfcrack02                                              | 68 |
| Gambar 4.53. Output Crash pada fuzzer-pdfcrack03                                              | 68 |
| Gambar 4.54. Proses triaging menggunakan AFLTriage                                            | 69 |
| Gambar 4.55. Output dari Proses Triaging                                                      | 69 |
| Gambar 4.56. Proses pemindahan file crash pada satu folder                                    | 70 |
| Gambar 4.57. Hasil crash pada file 2f6f361ada1210222eb5a429843f7711                           | 70 |
| Gambar 4.58. Hasil crash pada file 3627c4ae85bf591e25aa07a0da44c010                           | 71 |
| Gambar 4.59. Hasil crash pada file 6e65cbe014e74ee428f417ed43a94579                           | 71 |
| Gambar 4.60. Hasil crash pada file b4f07d137cf0732df4dd033e6b3bda90                           | 72 |
| Gambar 4.61. Hasil crash pada file bc89abf7925ac9e958b827106b821bea                           | 73 |
| Gambar 4.62. Hasil crash pada file ef5ebedac15e250aa50517b1a81656d5                           | 74 |
| Gambar 4.63. Stack Trace ASAN                                                                 | 74 |
| Gambar 4.64. Fungsi objStringToByte                                                           | 75 |
| Gambar 4.65. Fungsi parseRegulerString                                                        | 76 |
| Gambar 4.66. Ukuran buffer yang telah terdefinisi                                             | 77 |

Gambar 4.67 Proses Sanitasi Input ukuran Buffer

77

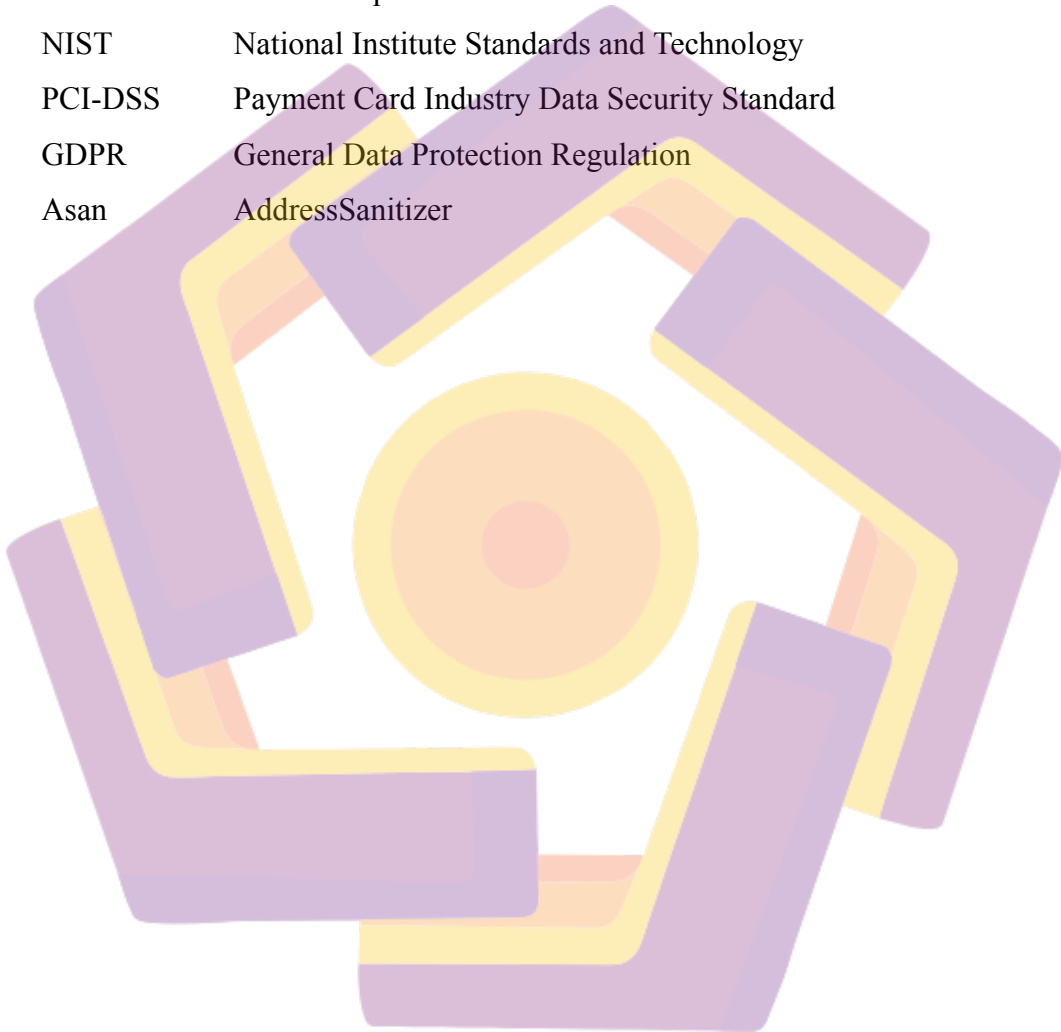
Gambar 4.68 Validasi Perangkat Lunak yang telah diperbaiki dengan menjalankan file uniq crash

77



## DAFTAR LAMBANG DAN SINGKATAN

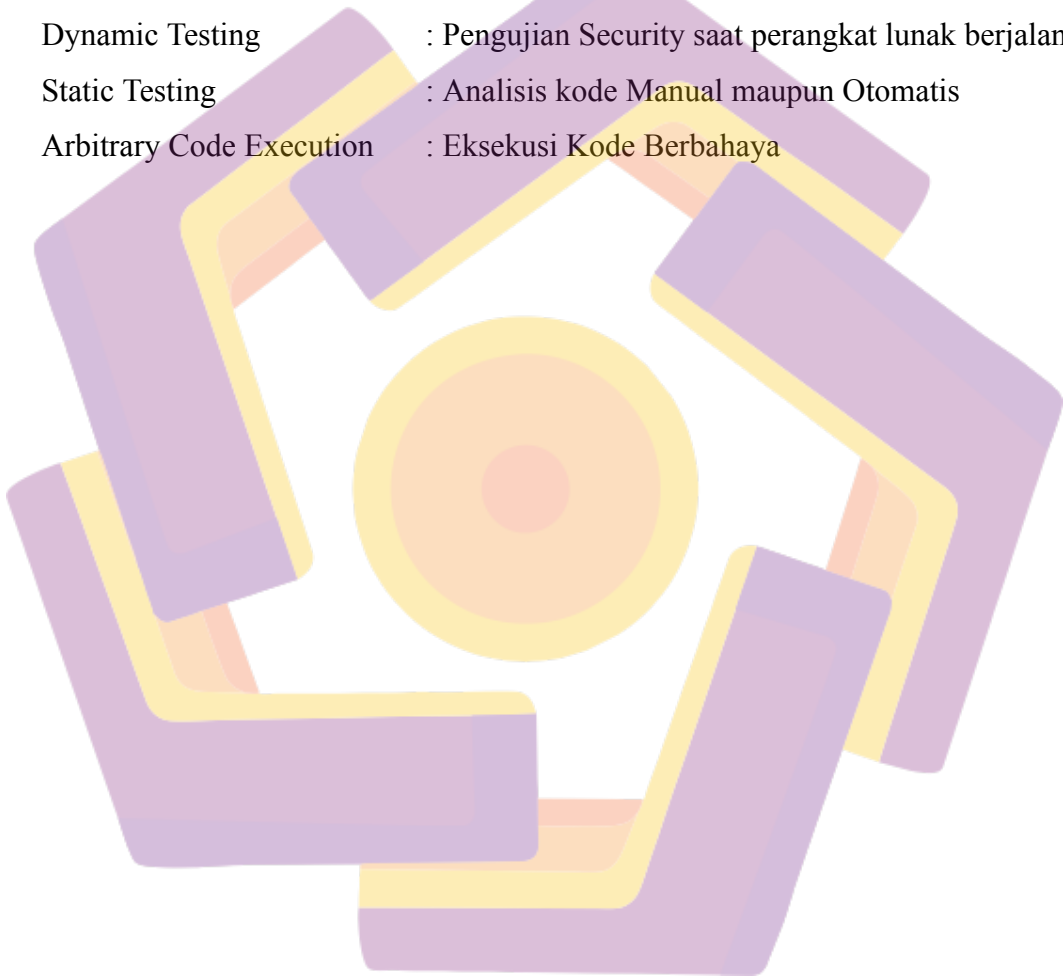
|         |                                              |
|---------|----------------------------------------------|
| AFL++   | American Fuzzy Looping                       |
| CIA     | Confidentiality, Integrity, Availability     |
| GCC     | GNU Compiler Collection                      |
| NIST    | National Institute Standards and Technology  |
| PCI-DSS | Payment Card Industry Data Security Standard |
| GDPR    | General Data Protection Regulation           |
| Asan    | AddressSanitizer                             |





## DAFTAR ISTILAH

|                          |                                                    |
|--------------------------|----------------------------------------------------|
| Fuzzing                  | : Inputan secara Acak                              |
| AddressSanitizer         | : Deteksi error memori saat fuzzing                |
| Buffer Overflow          | : Input yang melewati batas Buffer Heap atau Stack |
| Integer Overflow         | : Angka yang melampaui batas                       |
| Dynamic Testing          | : Pengujian Security saat perangkat lunak berjalan |
| Static Testing           | : Analisis kode Manual maupun Otomatis             |
| Arbitrary Code Execution | : Eksekusi Kode Berbahaya                          |



## INTISARI

Keamanan perangkat lunak merupakan aspek yang sangat krusial dalam pengembangan perangkat lunak modern, terutama dengan semakin meningkatnya ketergantungan terhadap teknologi digital dalam berbagai sektor kehidupan. Kerentanan dalam perangkat lunak dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk meluncurkan serangan yang mengancam prinsip-prinsip dasar keamanan informasi, yaitu kerahasiaan (Confidentiality), integritas (Integrity), dan ketersediaan (Availability). Penelitian ini bertujuan untuk mengetahui jumlah crash yang dihasilkan selama proses fuzzing, mengidentifikasi kerentanan keamanan dari crash yang ditemukan, serta mengevaluasi apakah seluruh crash tersebut benar-benar mengindikasikan adanya kerentanan atau hanya menyebabkan gangguan tanpa risiko eksploitasi. Fuzzing dilakukan pada sistem operasi Linux dengan perangkat lunak target yang telah dikompilasi menggunakan Instrumentasi dan Address Sanitizer (ASan). Hasilnya menunjukkan bahwa ditemukan sebanyak 76 file crash, di mana 6 di antaranya bersifat unik. Berdasarkan analisis, diketahui bahwa kerentanan yang ditemukan termasuk dalam kategori Stack Buffer Overflow. Kondisi ini membuka kemungkinan terjadinya Arbitrary Code Execution, terutama jika aplikasi dijalankan dengan hak akses tinggi. Penelitian ini menunjukkan bahwa sanitasi input yang ketat perlu diterapkan untuk mencegah eksploitasi. Diperlukan juga evaluasi menyeluruh terhadap fungsi-fungsi yang menangani input agar aplikasi lebih tahan terhadap serangan.

**Kata kunci:** Binary Fuzzing, American Fuzzy Looping, Buffer Overflow, AddressSanitizer, Keamanan Aplikasi.

## ***ABSTRACT***

*Software security is a crucial aspect of modern software development, especially with the increasing reliance on digital technology in various sectors of life. Vulnerabilities in software can be exploited by irresponsible parties to launch attacks that threaten the basic principles of information security, namely confidentiality, integrity, and availability. This research aims to determine the number of crashes generated during the fuzzing process, identify security vulnerabilities from the crashes found, and evaluate whether all crashes actually indicate a vulnerability or only cause disruption without the risk of exploitation. Fuzzing was performed on a Linux operating system with target software that had been compiled using Instrumentation and Address Sanitizer (ASan). The results showed that a total of 76 crash files were found, of which 6 were unique. Based on the analysis, it is known that the vulnerabilities found fall into the Stack Buffer Overflow category. This condition opens the possibility of Arbitrary Code Execution, especially if the application is run with high access rights. This research shows that strict input sanitisation needs to be implemented to prevent exploitation. A thorough evaluation of the functions that handle inputs is also needed to make the application more resilient to attacks.*

**Keyword:** Binary Fuzzing, American Fuzzy Looping, Buffer Overflow, AddressSanitizer, Software Security.